Is No-Ops the future of DevOps?
Who am I and why I care?

Peter Hall
DevOps / Platform Engineering Lead at DigIO

Systems Administrator → Systems Engineer → Operations Manager → Consultant
Definitions

**DevOps**
IT processes and practices that enable organisational value stream.

**Operations**
Engineers who specialise in the run time concerns of software / IT products.

**No-Ops**
Teams and organisations operating software / IT products without the support of Operations engineers.
**Current trends in IT infrastructure**

**IaaS has won**
Most companies are either leveraging a public cloud service or have a strategy to move in that direction.

**Kubernetes becoming industry standard**
All three major public cloud vendors have a Kubernetes solution. Only graduated CNCF project. Built-in support within Docker project.

**Serverless and PaaS increasing**
Consistently increasing interest in Serverless compute since launch of AWS Lambda. Multiple options to run containers / code without managing servers now.

**More SaaS options**
Good SaaS options available for almost all infrastructure services and every increasing application services too (i.e. Auth, API management, customer notifications).
Serverless architecture examples
Operations tasks... in decline

Physical server / storage / data centre management
- Public cloud is now the default choice

Configuration Management
- Cloud infrastructure provides the ability for immutable servers

Security Patching
- Security updates baked into CI/CD pipelines

Backup / Restore
- Still exists, but generally automated by the platform

Bespoke & Expensive CI/CD Pipelines
- Good SaaS and PaaS options to get pipelines created quickly
No-Ops is the future

Developers can create their own Serverless infrastructure in code – case closed!
But what about...

- Capacity planning
- SLO / SLA management
- Performance and resilience engineering
- Disaster recovery
- Incident response and post-mortem
- Cost management
Full stack engineers

Exists, but rare
How do you scale if it’s rare to find a wide range of skills in individual engineers

Backend + Frontend Dev
Even rarer for full stack engineer to be experienced with Operations

T-shaped engineers
Most common, wide range of experience but deep in one or two areas only
With minimum responsibility...

Comes minimum power...

Fast to get started

End up fighting the solution when you need something ‘custom’
Size matters

- 5 person startup haven't needed dedicated ops for a while
- Big bank unlikely to go completely serverless anytime soon
- Need for specialization grows with company and infrastructure size
The future Operations Engineer

Service Experts
Experts in what is available and how best to use it.

Platform Engineering
Turning the available services into an IT platform the business can leverage.

Resilience Engineering
or ‘chaos engineering’, building and testing infrastructure that expects and survives failure.

Performance Engineering
Knowing how to build high performing applications and limitations / opportunities in underlying services.

Data Protection
Ensuring your data stays private.

Cost Management
Serverless changes how we account for and build applications for cost efficiency.
What does it mean when they do
What impact will it have on the business when a service fails? What are our near and long term alternatives?

Know the options
Know what service options are out there and when best to use one over the other.

How to glue them together
How do we best integrate these services together and build architectures that provide maximum value for the business.

How they break
What sort of failure modes are built into the services you use?

What does it mean when they do
What impact will it have on the business when a service fails? What are our near and long term alternatives?
Final thoughts

Do you need to find a new career?
No! ... and we're hiring.

Do you need to keep embracing change?
Yes! Stop building and running what good quality SaaS can provide!

Be experts in how businesses can get value from their IT investment